

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application. No amendments are made to the claims, which are included herewith for completeness.

Listing of Claims

1. (Previously Presented) A computer-implemented system for retrieval and processing of a data set from one or more data sources comprising:

a query structure assembly module for defining a query structure based upon a plurality of query assembly rules and a desired data set, the query assembly rules being used by the query structure assembly module to evaluate the desired data set;

a syntax assembly module for defining at least one query language statement based upon the defined query structure;

a process optimization module for evaluating processing options based upon a database schema associated with the data source, the process optimization module including an intermediate data processing method module for evaluating a plurality of methods for generating intermediate data sets within the data source(s); and

whereby at least one query language statement is assembled and run against the data source(s) to return the desired data result set.

2. (Original) The system of claim 1, wherein the process optimization module includes a table selection module for evaluating the size of a selected set of tables from the database schema.

3. (Original) The system of claim 1, wherein the process optimization module includes an intermediate data processing selection module for evaluating the reusability of an intermediate data set in returning the defined data result set.

4. (Canceled)

5. (Original) The system of claim 1, wherein the process optimization module includes a join path selection module for evaluating the length of at least one join path used in returning the defined data result set.

6. (Original) The system of claim 1, wherein the query structure assembly module accesses a query assembly rule associated with a selected database management system.

7. (Original) The system of claim 1, wherein the syntax assembly module accesses a syntax description associated with a selected database management system.

8. (Original) The system of claim 1, wherein the process optimization module accesses a query assembly rule, a syntax description, or a syntax pattern associated with a selected database management system.

9. (Original) The system of claim 1, wherein the system is a component in an online analytical processing systems, a reporting system, a business intelligence system, or a data mining system.

10. (Previously Presented) A computer-implemented method of generating a query language statement to be run against one or more data sources, comprising the steps of:

generating a query structure based upon a database schema associated with the data source, query assembly rules, and a desired data result set, the query assembly rules being used to evaluate the desired data set;

generating query language syntax based upon the query structure for returning the desired data result set from the data source(s);

evaluating a plurality of query assembly rules, syntax descriptions, or syntax patterns for process optimization; and

evaluating a plurality of methods for generating intermediate data sets.

11. (Original) The method of claim 10, wherein the step of evaluating the plurality of query assembly rules, syntax descriptions, or syntax patterns for process optimization includes evaluating the size of a plurality of sets of identified tables for returning the desired data result set.

12. (Original) The method of claim 10, wherein the step of evaluating the plurality of query assembly rules, syntax descriptions, or syntax patterns for process optimization includes evaluating the reusability of intermediate data sets.

13. (Canceled)

14. (Original) The method of claim 10, wherein the step of evaluating a plurality of query assembly rules, syntax descriptions, or syntax patterns for process optimization includes evaluating a plurality of join paths used in returning the desired data result set.

15. (Original) The method of claim 10, wherein the step of evaluating a plurality of query assembly rules, syntax descriptions, or syntax patterns for process optimization includes evaluating at least one query assembly rule, syntax description, or syntax pattern associated with a selected database management system.

16. (Original) The method of claim 10, wherein the step of generating a query structure includes evaluating at least one query assembly rule associated with a selected database management system.

17. (Original) The method of claim 10, wherein the step of generating query language syntax includes evaluating at least one syntax description or syntax pattern associated with a selected database management system.

18. (Original) The method of claim 10, wherein the method is implemented in an online analytical processing systems, a reporting system, a business intelligence system, or a data mining system.

19. (Previously Presented) A medium having a processor readable program code embodied therein for retrieving and processing data from one or more data sources comprising:

code for causing the processor to evaluate a plurality of sets of tables within the data source(s) for generating a desired data result set;

code for causing the processor to evaluate at least one intermediate data set for reusability in generating the desired data result set;

code for causing the processor to evaluate a plurality of methods for generating intermediate data sets for use in generating the desired data result set;

code for causing the processor to evaluate a plurality of join paths used for joining tables to return the desired data result set; and

code for causing the processor to assemble at least one query language statement based upon the query structure and the evaluations of the plurality of sets of tables, the at least one intermediate data set, the plurality of methods for generating intermediate data sets, and the plurality of join paths.

20. (Original) The medium of claim 19, further comprising code for causing the processor to evaluate at least one query assembly rule associated with a selected database management system.

21. (Previously Presented) A computer-implemented system for constructing a structured query language statement to be run against at least one database, comprising:

a query structure assembly module for constructing a query structure based upon an evaluation of a desired data set by at least one query assembly rule;

a syntax assembly module for defining at least one query language statement based upon the constructed query structure; and

a process optimization module for evaluating the construction of the query structure and the defining of the at least one query language statement, the evaluation occurring during the construction of the query structure and/or during the defining of the at least one query statement, the process optimization module serving to evaluate a plurality of methods for generating intermediate data sets.

22-25. (Cancelled)

26. (Previously Presented) A computer-implemented system for retrieval and processing of a data set from one or more data sources comprising:

a query structure assembly module for defining a query structure based upon a plurality of query assembly rules and a desired data set, the query assembly rules being used by the query structure assembly module to evaluate the desired data set;

a syntax assembly module for defining at least one query language statement based upon the defined query structure;

a process optimization module for evaluating processing options based upon a database schema associated with the data source, the process optimization module including an intermediate data processing method module for evaluating a plurality of methods for generating intermediate data sets within the data source(s);

whereby at least one query language statement is assembled and run against the data source(s) to return the desired data set; and

wherein the intermediate data processing method module determines whether creation of a permanent table, temporary table, view, derived table, or sub-query is the most efficient method for handling intermediate data calculations.

27. (Previously Presented) A computer-implemented method of generating a query language statement to be run against one or more data sources, comprising the steps of:

generating a query structure based upon a database schema associated with the data source, query assembly rules, and a desired data result set, the query assembly rules being used to evaluate the desired data set;

generating query language syntax based upon the query structure for returning the desired data result set from the data source(s);

evaluating a plurality of query assembly rules, syntax descriptions, or syntax patterns for process optimization;

evaluating a plurality of methods for generating intermediate data sets; and

wherein the step of evaluating a plurality of methods for generating intermediate data sets comprises determining whether creation of a permanent table, temporary table, view, derived table, or sub-query is the most efficient method for handling intermediate data calculations.

28. (Previously Presented) A medium having a processor readable program code embodied therein for retrieving and processing data from one or more data sources comprising:

code for causing the processor to evaluate a plurality of sets of tables within the data source(s) for generating a desired data result set;

code for causing the processor to evaluate at least one intermediate data set for reusability in generating the desired data result set;

code for causing the processor to evaluate a plurality of methods for generating intermediate data sets for use in generating the desired data result set;

code for causing the processor to evaluate a plurality of join paths used for joining tables to return the desired data result set;

code for causing the processor to assemble at least one query language statement based upon the query structure and the evaluations of the plurality of sets of tables, the at least one intermediate data set, the plurality of methods for generating intermediate data sets, and the plurality of join paths; and

code for determining whether creation of a permanent table, temporary table, view, derived table, or sub-query is the most efficient method for handling intermediate data calculations.

29. (Previously Presented) A computer-implemented system for constructing a structured query language statement to be run against at least one database, comprising:

a query structure assembly module for constructing a query structure based upon an evaluation of a desired data set by at least one query assembly rule;

a syntax assembly module for defining at least one query language statement based upon the constructed query structure;

a process optimization module for evaluating the construction of the query structure and the defining of the at least one query language statement, the evaluation occurring during the construction of the query structure and/or during the defining of the at least one query statement, the process optimization module serving to evaluate a plurality of methods for generating intermediate data sets; and

wherein the process optimization module's evaluation of a plurality of methods for generating intermediate data sets comprises determining whether creation of a permanent table, temporary table, view, derived table, or sub-query is the most efficient method for handling intermediate data calculations.